4W－7

The Internationalized Environment Enabling Proper Text Processing

Kazutomo Uezono†, Tomoko Kataoka*, Dawa Idemucao†,
Yutaka Kataoka*, and Hiroyoshi Ohara†

† School of Science and Engineering, Waseda University * Media Network Center, Waseda University

1. Introduction

Since computer network has covered almost all over the world, the occasion to communicate using various languages/scripts simultaneously is increased, which requires to process all the scripts correctly on computers, namely, *Internationalized Computing Environment* (ICE). However, most of the computing environment cannot process all the scripts correctly and simultaneously, because they are based on *POSIX Locale Model* [1] or *limited multilingual models* which partly support ISO 2022 [3], that is *Localized Computing Environment* (LCE).

On the LCE, especially on POSIX Locale Model, all texts are processed with each *language-specific information* called *Locale*. In case of mixing arbitrary combination of scripts, users should change the Locale interactively. But Locale cannot be changed because of the ambiguous POSIX specifications which do not have further interpretations. This shows that the set of Locales does not establish ICE. Therefore, the ICE should be constructed without Locales and should not depend on each language-specific information.

Text processing have been considered as orthography and/or language dependent. But, by the analyses of all scripts of the world, a *Character* was defined independently from language, orthography, and glyph, and the relations among Character, glyph, and language were also defined. Thus, it was impossible to clarify the relation between ICE and LCE, and the ICE was realized with keeping backward compatibilities for the LCE including POSIX Locale Model.

2. Definition of a Character

Scripts in the world can be generally classified into the following categories; 1) *Phonemic*, 2-1) *Conjunct Syllabic*, 2-2) *Pure Syllabic* and 3) *Ideogrammic*. This classification only points the object directed by the character, which does not relate with text processing. Therefore, from a construction point of view, the scripts was classified into a) *Non-Conjunctive* and b) *Conjunctive* (Fig. 1) [5].

Since a conjunctive script has a set of symbols, a character of the conjunctive script is constructed from selected symbols by the *construction rule*. On the other hand, a non-conjunctive script also has a set of symbols, but a character of the non-conjunctive script is constructed without the rule because each symbol is a character. However, the latter can be considered to be constructed with the *empty construction rule*. By the above, a character was defined as a set of symbols constructed by the construction rule (Fig. 2).

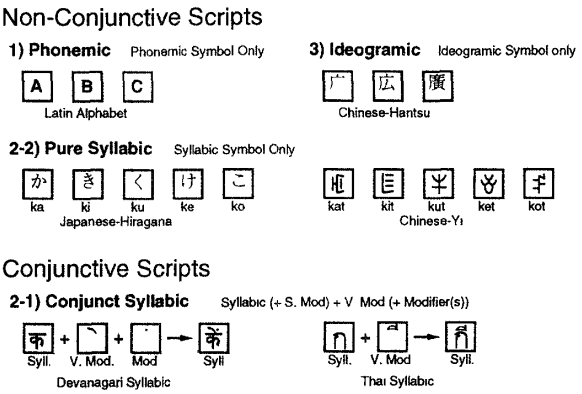

Figure 1. Classification of Scripts

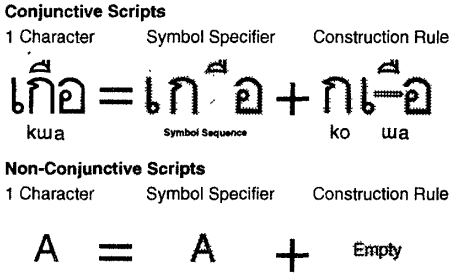

Figure 2. Definition of a Character

Since the construction rule inheres in a script, it is not influenced by a language, which shows that a character is constructed without language information.

3. Relation between Character and Glyph

A glyph of a character is changed according to its writing direction, position in a word and line progress. Therefore, a character, which specifies a set of glyphs, can be defined as a name of a set of glyphs (Fig. 3).

By the above definition, the function Drawing can be defined as selecting a suitable element in a set of glyphs specified by a character, and arranging according to the writing conditions (writing direction, line progress and

so on). Thus, the function Drawing is independent from language information.

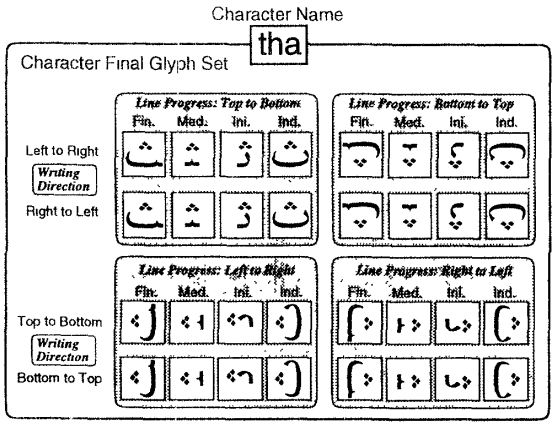

Figure 3. A Character Specifies a Glyph Set

4. Construction of ICE

In general, texts are read and written based on a character. Thus, a character was defined as a unit of text processing. Since a character and its glyph are determined without language dependency, text processing can be classified into 1) *language-independent processing* (ex. Insertion, Deletion, Search and Line Breaking) and 2) *language-dependent processing* (ex. Spell Checking).

As Type 1) is processed without language dependency, Computing Environment for the type 1) is constructed without Locale dependency. Therefore, the ICE can be realized with type 1). Note that *Inner process code* in the ICE should be uniquely defined to mix all texts. Type 2) can be processed by adding language information to the ICE, which enables to realize *the Multilingual Computing Environment* based on the ICE.

Though the LCE depends on the Locale, its realization can be considered to give the ICE default status, i.e., limitations of its use. As a result, the LCE was defined as a partial set of the ICE (Fig. 4).

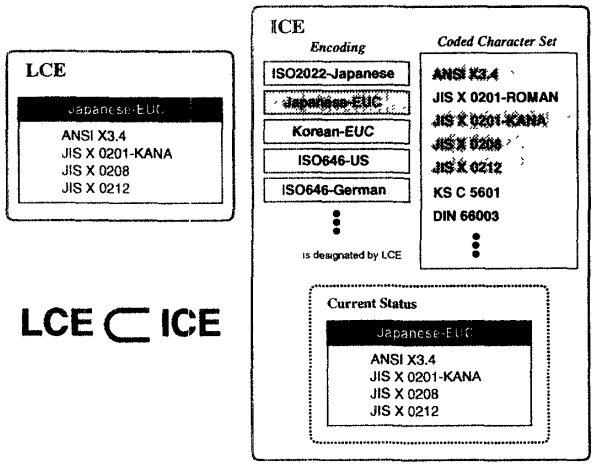

Figure 4. Relation between the ICE and the LCE

5. Concluding Remarks

It was clarified that the ICE was constructed as a superset of the LCE without language dependency. Once the set of process unit is defined, mixed texts of all the scripts in the world can be processed correctly under the ICE.

To realize the ICE, *Global IOTMC Model* and *Multi-locale Model* were developed [6]. The Global IOTMC Model were realized with supporting not only ISO 646 [2], ISO 4873 [4] and ISO 2022 but also other International/National specifications and code extensions derived from analyses of scripts, languages and orthographies. On the other hand, the Multi-locale Model was realized to support various encodings for keeping interoperability.

The multilingual system named the *System 1* was developed based on the Global IOTMC Model, which provides to enhance *Operating systems* and *Windowing systems* with replacing Locale-specific functions. It contributes to easily develop the multilingual utilities and applications (Fig. 5).

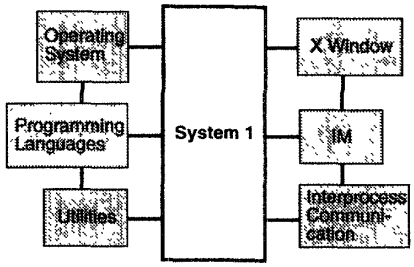

Figure 5. The System 1 Environment

References

[1] ISO/IEC 9945-1: 1990, Information technology – Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language].

[2] ISO/IEC 646: 1991, Information technology – ISO 7-bit coded character set for information interchange.

[3] ISO/IEC 2022: 1994, Information technology – Character code structure and extension techniques.

[4] ISO/IEC 4873: 1991, Information technology – ISO 8-bit code for information interchange – Structure and rules for implementation.

[5] Kataoka, Y., et al., Multilingual Computing of All Characters in the World, Symposium of Humanities and Computers '96, October 1996, pp 97-108, Research fund of The Ministry of Education.

[6] Uezono, K., et al., The Multilingual Text Processing (5): Extension of POSIX to Multilingual Processing, Proceedings of the 53th General Meeting of IPSJ, Vol. 4, September 1996, pp 131-132.